**Soutenance**

Bonjour à tous et bienvenue à cette présentation. Nous allons donc aujourd’hui discuter de la solution technique retenue pour mener à bien le projet Menu Maker porté par Qwenta. Notre présentation sera structurée de la manière suivante : nous commencerons par présenter le projet, son fonctionnement et son but ; puis nous expliquerons le travail de veille que nous avons effectué pour aboutir aux solutions que nous vous présenterons ensuite ; enfin nous expliciterons la manière dont sera géré le projet, en termes de temporalité, d’organisation et d’objectifs.

Je vous propose de commencer dès maintenant en vous présentant le projet. Le but de Menu Maker est le suivant : «  Menu Maker est un site permettant aux restaurateurs d’afficher et de mettre en page leurs menus facilement, en quelques clics. ». Dans les faits, ce site remplira plusieurs fonctions que voici : pouvoir créer un menu qui comportera un nom, un prix, une description et des catégories de plat ; pouvoir personnaliser son menu, notamment avec un branding restaurateur, la personnalisation du logo, des couleurs de base et de la police utilisée ; pouvoir diffuser son menu via divers canaux que sont Deliveroo, Instagram et l’exportation au format PDF ; enfin l’impression de menu directement depuis le site.

Pour comprendre un peu mieux le fonctionnement du site, voici l’arborescence de celui-ci. Nous avons donc au départ l’accueil du site (aussi appelé landing page), puis 3 onglets : le premier dédié aux tarifs, le second à l’espace utilisateur (ou dashboard) et le dernier à des exemples de menus. Depuis l’espace utilisateur nous avons ainsi accès à la création de menus, à nos menus précédemment créés et à nos informations personnelles.

Voici donc le design retenu pour le site. Nous avons ici notre page d’accueil avec nos 3 onglets disponibles en haut à droite.

Du côté des spécifications techniques souhaitées pour le site, il devra être conçu pour ordinateur uniquement, il n’y a pas de version mobile prévue. Sa charte graphique est également définie : fond beige et blanc, textes en noir, boutons en vert. Pour les navigateurs, nous visons une compatibilité pour Chrome, Firefox et Safari. Le nom de domaine est en cours de validation mais sera très certainement un sous domaine de Qwenta. Pour finir, le site doit être navigable depuis le clavier et lisible par un lecteur d’écran.

Le projet bien cerné, nous avons pu passer au travail de veille, et ce grâce à l’outil Feedly. Feedly est un outil proposant à ses utilisateurs d’organiser leur veille d’information en choisissant les médias qu’ils souhaitent suivre et quel contenu ils veulent lire classé par catégorie. Dans la veille informationnelle, il existe deux grandes méthodes pour la mener à bien : la méthode « pull », l’approche la plus classique, l’utilisateur se rend directement et régulièrement sur internet pour en « tirer » les informations les plus récentes dans un domaine particulier en utilisant des moteurs de recherche, des métamoteurs, des annuaires, des signets de pages web à revisiter ou des pages de liens ; la méthode « push », où l'information est « poussée » de manière automatique vers l’utilisateur en fonction de ses préférences et de ses critères en utilisant des flux RSS , des alertes par courriel, des listes de diffusion, des logiciels de surveillance de pages web ou des abonnements à un centre de documentation. Avec Feedly, il est possible d’utiliser les deux méthodes, c’est ce que nous avons fait afin d’avoir la veille la plus efficace possible.

Ici, vous pouvez voir notre veille « push », avec à gauche nos différents flux RSS (ou Feeds) classés par catégories afin de ne pas s’éparpiller, à côté nous avons l’exemple de notre Feed dédié au Front-End, avec les derniers articles publiés qui s’ajoutent automatiquement et enfin à droite nous avons l’outil de recherche permettant de rechercher des flux RSS et de s’y abonner. On constate qu’il est même possible d’en rechercher sur différents réseaux sociaux.

Abordons maintenant les choix techniques et technologiques que nous avons pu retenir à l’issue de notre veille. Pour commencer, pour la partie Front-End, nous avons décidé de développer sous React, qui est une bibliothèque JavaScript utilisée pour construire des interfaces utilisateur. Les avantages de React sont nombreux, notamment sa facilité de prise en main, ses performances et sa flexibilité exceptionnelles, et ses composants réutilisables. Côté Back-End, nous avons retenu NodeJS, qui est une plateforme logicielle permettant d’exécuter du JavaScript côté serveur. L'un des avantages de NodeJS est la possibilité d'exécuter plusieurs requêtes vers le serveur simultanément, il n’y a donc pas besoin d’attendre qu’une requête lancée précédemment soit terminée. NodeJS est également très performant et flexible, ce qui en fait donc un choix judicieux pour notre projet. Enfin, pour ce qui est de la partie base de données, nous avons sélectionné mySQL, qui est un système de gestion de bases de données relationnelles. mySQL est très réputé et parfaitement adapté à notre projet de par sa simplicité d’utilisation et de configuration.

Concernant les différentes fonctionnalités du site, nous avons fait les choix suivants : pour l’authentification utilisateur, nous utiliserons Nodemailer (librairie Nodejs) pour envoyer un mail de confirmation à l’utilisateur, et localStorage (objet de stockage web) pour gérer la session de l’utilisateur. Nous utiliserons les librairies React-pdf pour l’exportation en PDF et React-to-print pour l’impression de menus. Nous utiliserons également Redux pour la partie personnalisation de menus. Enfin, nous ferons appel aux API de Deliveroo et Instagram pour les fonctionnalités d’exportation et de partage vers ces plateformes.

Ensuite, pour la mise en forme du site, nous nous servirons des librairies React-modal pour les fenêtres modales, React Hook Form pour les différents formulaires, et enfin les composant React Card pour le dashboard et React Tabs pour l’onglet « tarifs ».

D’autre part, nous avons jugé utile d’intégrer plusieurs services tiers, tels que : Google Analytics pour améliorer le référencement du site et analyser son audience ; Cloudflare afin d’augmenter les performances et la sécurité du site.

Pour finir, nos recommandations en termes de sécurité sont les suivantes : la mise en place d’un certificat SSL pour sécuriser la connexion ; des sauvegardes régulières du site ; des mots de passes renforcés (recommandations CNIL) ; la mise en place de restrictions d’accès aux fichiers sensibles. Nous prévoyons également le déploiement de correctifs et de mises à jour régulières pour une maintenance optimale du site.

Il est temps d’aborder la gestion du projet en lui-même. Pour le mener à bien, nous avons prévu l’équipe suivante : 1 product owner qui fera le lien entre le client et les équipes, en tenant informé chaque partie des avancements, changements et imprévus du projet ; 1 développeur Front-End pour développer la partie vitrine du site ; 1 développeur Back-End pour développer les coulisses du site (connexion utilisateur par exemple) ; 1 développeur FullStack pour faciliter les interactions entre Front et Back et venir en renfort sur les tâches les plus ardues.

Côté méthodologie, nous avons retenu une méthodologie de développement agile, qui est une approche dont l’objectif est de distribuer en continu des logiciels opérationnels créés sur la base d’itérations rapides, aussi appelées sprints. Chaque sprint représente un cycle relativement court durant lequel, l’équipe va devoir réaliser une succession de tâches pour mener à bien une phase du projet. Pour un même projet, différents sprints se succèdent pour, à la fin, obtenir un produit fini. De cette manière, la prise de risque est amoindrie et l’avancée se fait par étape. Il est alors plus simple d’entreprendre des corrections au fur et à mesure de l’avancée du projet ou d’effectuer des modifications en cours de route, d’adapter les prochains sprints à de nouveaux besoins ou encore, de repenser le projet selon les obstacles rencontrés. Cela permet d’affiner la conformité du projet final avec les attentes du client et d’améliorer sa satisfaction.

Pour mettre en place cette méthodologie, nous nous appuierons sur un outil de gestion de projet, nommé Trello, sur lequel nous aurons notre tableau Kanban, qui nous permettra de visualiser l’avancée du projet

Le tableau se présente ainsi, nous avons nos cartes représentant les différentes tâches à effectuer et celles-ci sont classées dans différentes colonnes indiquant leur statut : « A faire », « En cours », « Terminé », « A tester ».

Pour le chiffrage du projet, nous aurons des sprints de 2 semaines, avec pour objectif l’implémentation d’1 à 2 fonctionnalités. Ainsi, étant donné la complexité relative des différentes tâches, nous prévoyons la livraison de toutes les fonctionnalités de priorité 1 d’ici 4 mois, la livraison de toutes les fonctionnalités de priorité 2 d’ici 7 mois, et la livraison des dernières fonctionnalités d’ici 8 mois.

Notre dernier point concerne la communication. Pour celle-ci, nous préconisons des réunions internes quotidiennes (de 15-30mn en distanciel) pour faire le point sur la situation, l’avancement et l’assignation des tâches ; des réunions hebdomadaires (de 15-30mn en distanciel) entre le client et le product owner afin de faire remonter les avancées du sprint en cours et les éventuels imprévus ; et des réunions bilan à chaque fin de sprint (de 30-45mn en présentiel) entre le client et l’ensemble de l’équipe pour présenter les réalisations accomplies et discuter des sprints à venir (implémentation de nouvelles fonctionnalités, modifications de fonctionnalités existantes, corrections de bugs, mises à jour…).

Voilà qui conclut cette présentation, je vous remercie de m’avoir écouté et serais heureux de répondre à vos questions.